

## Claims

I claim:

1. A scheduling method for scheduling tasks providing a system designed on an object-oriented basis by using a computer, comprising:

a first step of extracting classes forming said system from design information thereof, generating tasks corresponding to the extracted classes, and storing information on the generated tasks into a given storage device;

a second step of extracting dependency between the classes from the design information of said system, setting the dependency between the corresponding tasks on the basis of the dependency, and storing information on the set dependency between the tasks into the given storage device; and

a third step of scheduling the tasks on the basis of the information on the tasks stored in said given storage device and the information on the dependency between the tasks.

2. The method according to claim 1, wherein, if given classes have a relation in which a development of one class requires an earlier development of the other class in said second step, the dependency is set in such a way that a task corresponding to one class concerned is started after termination of a task corresponding to the other class concerned.

3. The method according to claim 1, wherein, if given classes have an interdependence relation in said second step, the dependency is set in such a way that each task is terminated after awaiting tasks corresponding to all the classes in the interdependence to be ready to terminate.

4. A scheduling method for scheduling tasks providing a system designed on an object-oriented basis by using a computer, comprising:

a first step of extracting classes forming said system from design information thereof, generating tasks corresponding to the extracted classes, and storing information on the generated tasks into a given storage device;

a second step of extracting complexity of the classes from the design information of said system, estimating workload required for implementation of the classes on the basis of the complexity, and storing a result of the workload estimation into the given storage device; and

a third step of scheduling the tasks on the basis of the information on the tasks stored in said given storage device and the result of the workload estimation.

5. The method according to claim 4, wherein, in said second step, past workload achievements in each class are used to estimate the workload of the classes in a multi-regression analysis.

6. A program product for scheduling tasks providing a

system designed on an object-oriented basis by controlling a computer to enable the computer to execute:

a first process of extracting classes forming said system from design information thereof, generating tasks corresponding to the extracted classes, and storing information on the generated tasks into a given storage device;

a second process of extracting dependency between the classes from the design information of said system, setting the dependency between the corresponding tasks on the basis of the dependency, and storing information on the set dependency between the tasks into the given storage device; and

a third process of scheduling the tasks on the basis of the information on the tasks stored in said given storage device and the information on the dependency between the tasks.

7. The program product according to claim 6, wherein, if given classes have a relation in which a development of one class requires an earlier development of the other class in said second process, the dependency is set in such a way that a task corresponding to one class concerned is started after termination of a task corresponding to the other class concerned.

8. The program product according to claim 6, wherein, if given classes have an interdependence relation in said second process, the dependency is set in such a way that each task is terminated after awaiting tasks corresponding to all

the classes in the interdependence to be ready to terminate.

9. The program product according to claim 6,

wherein said computer is further caused to execute a fourth process of extracting complexity of said classes from said design information of the system, estimating workload required for implementation of the classes on the basis of the complexity, and storing a result of the workload estimation into the given storage device; and

wherein said third process includes scheduling the tasks with considering said result of the workload estimation in said fourth process.

10. The program product according to claim 9,

wherein, in said fourth process of said program, past workload achievements in each class are used to estimate the workload of the classes in a multi-regression analysis.

11. The program product according to claim 6, wherein

said computer is further caused to execute a fifth process of generating and outputting a Gantt chart having a description of said task schedule generated in said third process.

12. The program product according to claim 6, wherein

a class diagram described in UML™ is used as said system design information in said first and second processes.

13. A task scheduling apparatus, comprising:

a task generation unit for extracting classes forming a system from design information of the system designed on an object-oriented basis and generating tasks corresponding to the extracted classes;

a dependency setting unit for extracting the dependency of the classes from the design information of said system and setting dependency between the corresponding tasks on the basis of the dependency;

a workload estimation unit for extracting complexity of the classes from the design information of said system and estimating workload required for implementation of the classes on the basis of the complexity; and

a scheduling unit for scheduling the tasks on the basis of the information on the tasks generated by said task generation unit, information on the dependency between the tasks set by said dependency setting unit, and the estimation result made by said workload estimation unit.

14. The apparatus according to claim 13, wherein, if given classes have a relation in which a development of one class requires an earlier development of the other class, said dependency setting unit sets the dependency in such a way that a task corresponding to one class concerned is started after termination of a task corresponding to the other class concerned.

15. The apparatus according to claim 13, wherein, if given classes have an interdependence relation, said dependency setting unit sets the dependency in such a way

that each task is terminated after awaiting tasks corresponding to all the classes in the interdependence to be ready to terminate.

16. The apparatus according to claim 13, wherein said workload estimation unit uses past workload achievements in each class to estimate the workload of the classes in a multi-regression analysis.

17. The apparatus according to claim 13, wherein said scheduling unit describes said generated task schedule in a Gantt chart and outputs it.

18. The apparatus according to claim 13, wherein said system design information is a class diagram described in UML™.